

INSTITUTO FEDERAL
SANTA CATARINA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CAMPUS SÃO JOSÉ
ENGENHARIA DE TELECOMUNICAÇÕES
Disciplina: Sistemas Operacionais
Alunas: Marina Souza e Schaiana Sonaglio

TerminALL - Interpretador de Comandos

Descrição do Projeto:

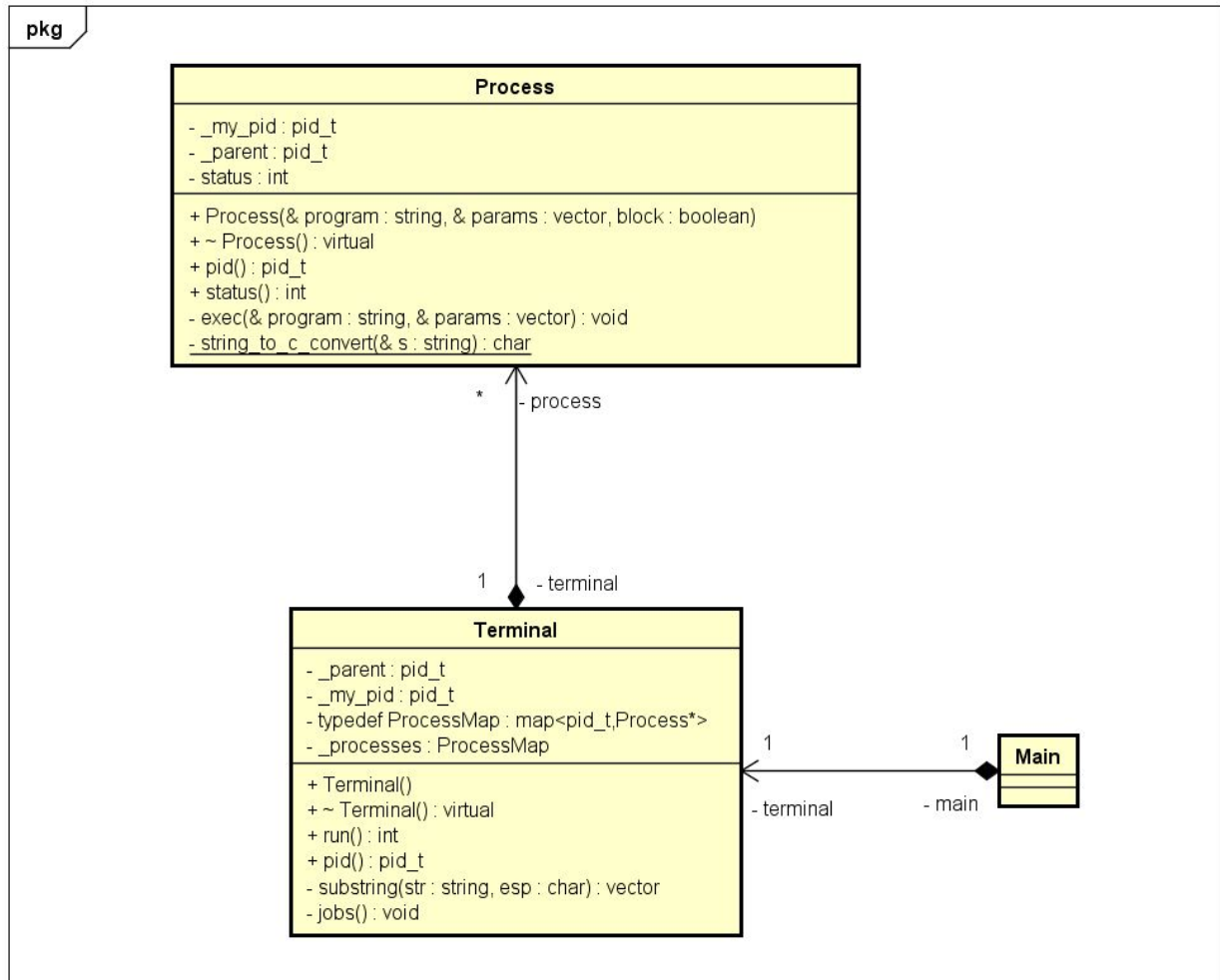
O objetivo do projeto é simular um terminal de execução do linux, que recebe um comando e cria um processo no sistema operacional para executá-lo.

Na implementação, o usuário digita o comando solicitado, o método “substring()” na classe Terminal divide a string recebida em um vetor de substrings. Cada indexação do vetor representará um argumento do comando a ser executado pelo terminal. Se primeira string digitada pelo usuário for “exit”, o programa encerra (independente de ter ou não espaços antes da palavra). Caso o comando solicitado pelo usuário seja diferente de “exit”, o programa testa se o comando de entrada possui “&”, caso tenha, o processo do terminal permanecerá em execução junto com o processo filho (método “exec()”); caso não tenha, será utilizado o “wait()” para bloquear a execução do pai, até o filho finalizar a sua execução.

O tratamento do comando é realizado pelo método “Process()”, na classe Process, que recebe como parâmetro a primeira palavra do comando, o vetor de substrings (que possui os parâmetros digitados pelo usuário no terminal) e uma variável booleana indicando se há “&” no comando (bool true) ou não (bool false).

Para verificar os processos em execução, o usuário pode digitar “jobs” no terminal, o método “jobs()” lista os processos em execução. Ele percorre o map _process com o iterador, buscando pelo ID do processo, depois disso, localiza o arquivo “status” na pasta PROC do Linux e, nele, coleta o nome do processo, seu ID e status.

Diagramas UML:



Métodos Utilizados:

Terminal::Terminal(): Construtor da classe Terminal. Inicializa o map `_process`, que irá armazenar os IDs dos processos.

Terminal::run(): Mostra o prompt de comando. Recebe o comando. Separa o comando em substrings, utilizando o método `substring()`. Encerra o TerminALL, caso o comando digitado seja "exit". Executa o método `jobs()`, caso o comando digitado seja "jobs". Busca e mostra o

nome do PC e do usuário que está executando o TerminALL. Cria um objeto da classe Process; se o comando for executado com “&” no final, armazena no map _process.

Terminal::jobs(): Percorre o map _process, busca pelo ID do processo, acessa a pasta de processos (PROC) e localiza o arquivo com as informações do mesmo. Coleta o nome do processo e seu status.

Terminal::substring(): Separa o comando em substrings, procurando pelo caractere vazio. Remove os caracteres vazios.

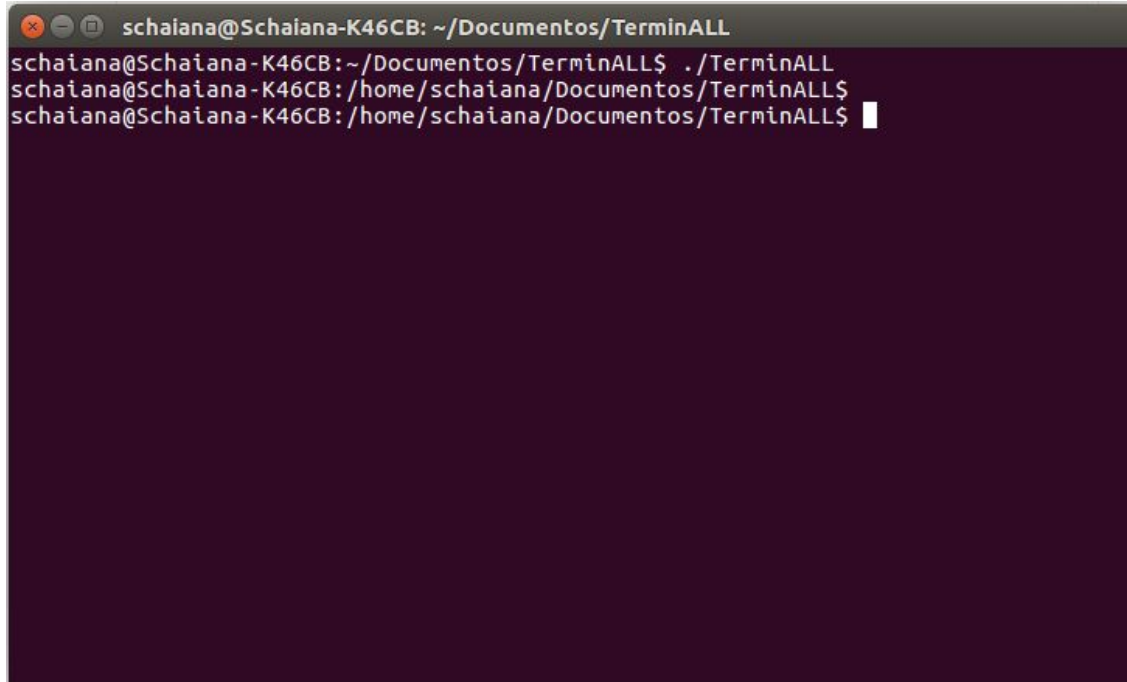
Process::Process(): Construtor da classe Process. Cria o processo filho, utilizando o fork(). Verifica se houve erro ao criar o processo. Verifica se é o processo pai. Armazena o ID do processo filho e do processo pai. Verifica se foi utilizado “&” depois do comando digitado no TerminALL; se não foi, bloqueia a execução do processo pai utilizando wait(). Verifica pelo ID do processo se é o processo filho, caso seja, executa o comando digitado utilizando o método exec().

Process::exec(): Converte o comando (programa e parâmetros) para vetor de char, após utilizar o método string_to_c_convert() para converter strings em char. Coloca NULL na última posição do vetor de vetor de char, conforme manual do exec.

Process::string_to_c_convert(): Converte string para char.

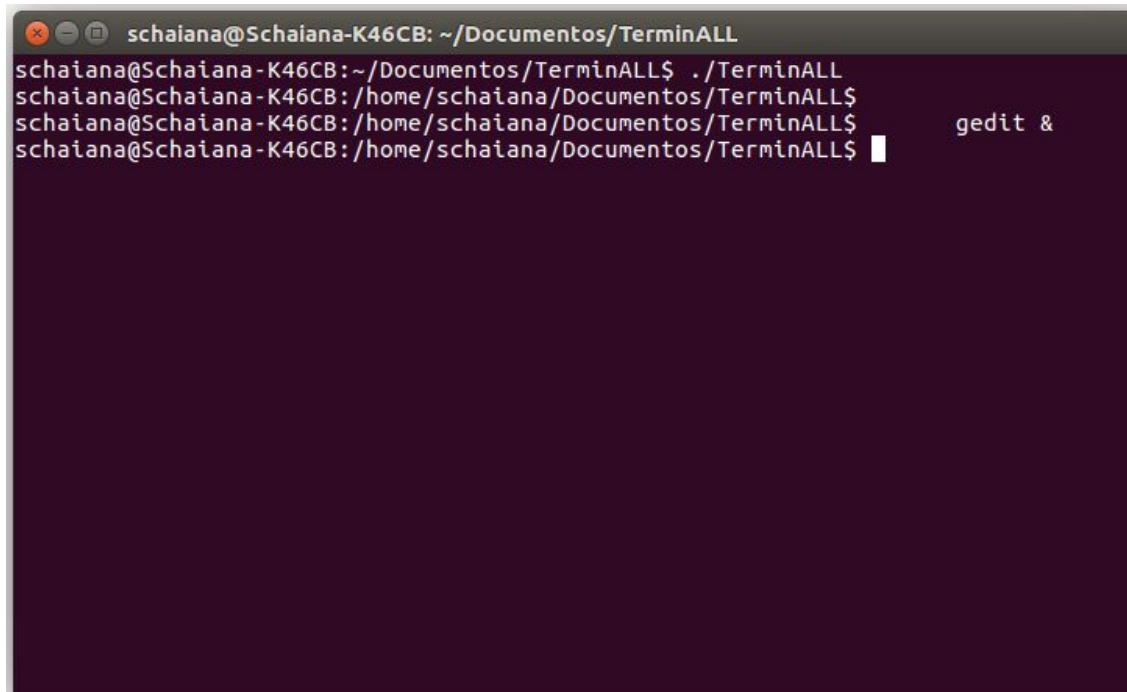
Testes Realizados:

Enter sem comando:



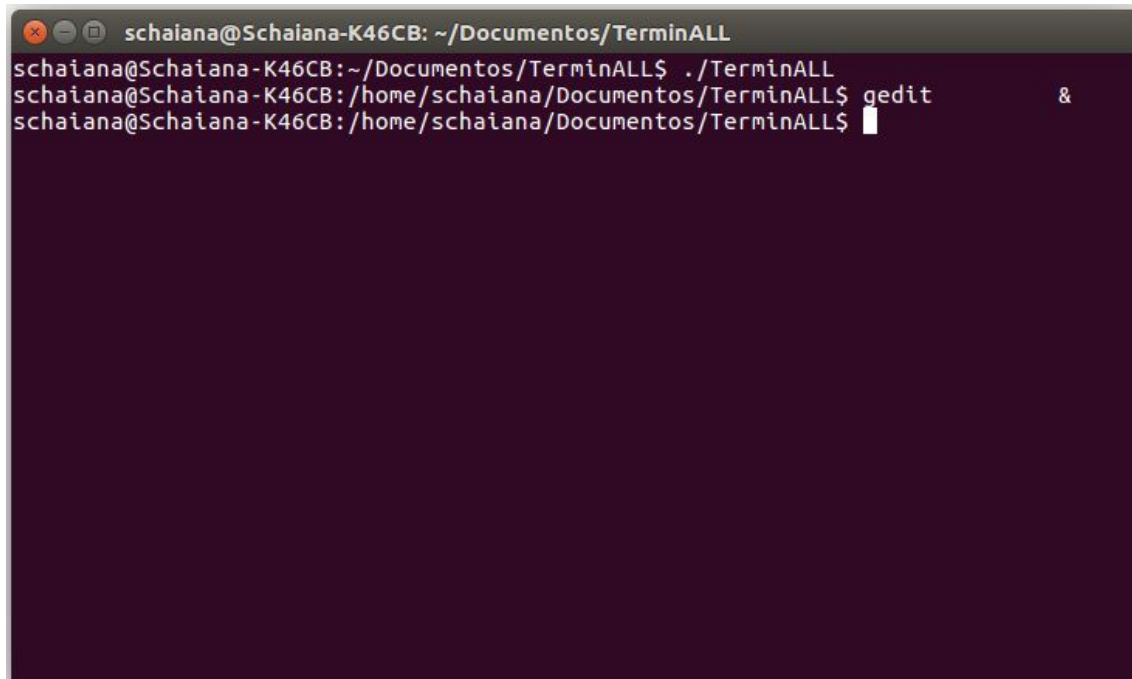
```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB: /home/schaiana/Documentos/TerminALL$
schaiana@Schaiana-K46CB: /home/schaiana/Documentos/TerminALL$
```

Comando com espaço antes:



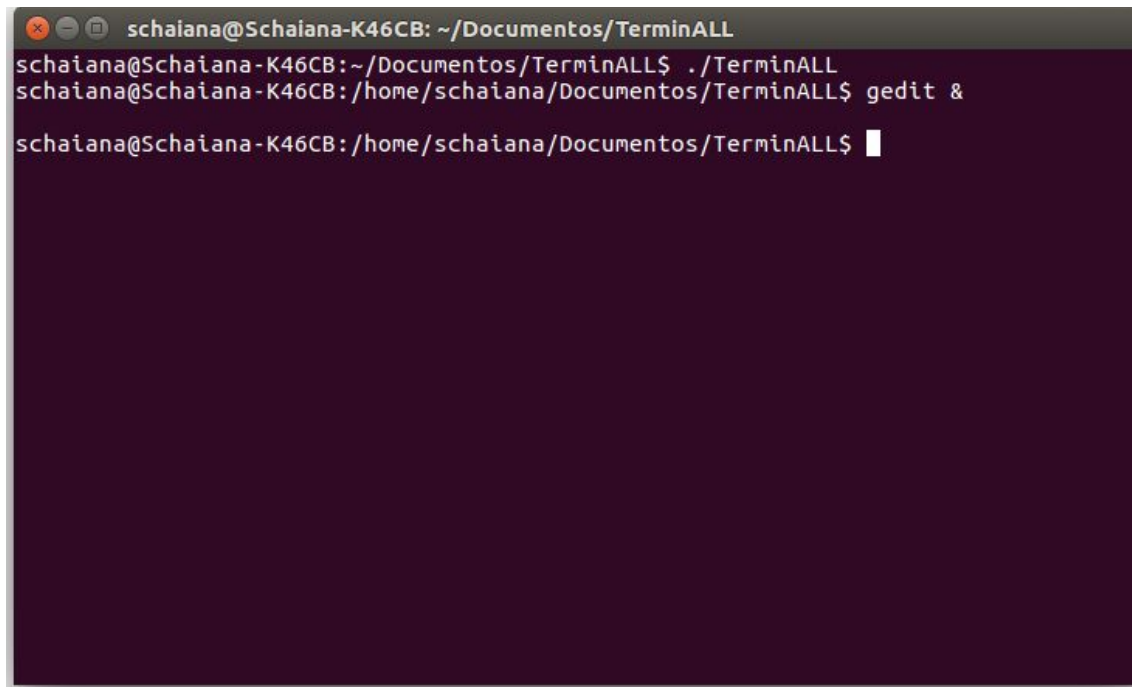
```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB: /home/schaiana/Documentos/TerminALL$
schaiana@Schaiana-K46CB: /home/schaiana/Documentos/TerminALL$ gedit &
schaiana@Schaiana-K46CB: /home/schaiana/Documentos/TerminALL$
```

Comando com espaço no meio:



```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ gedit      &
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$
```

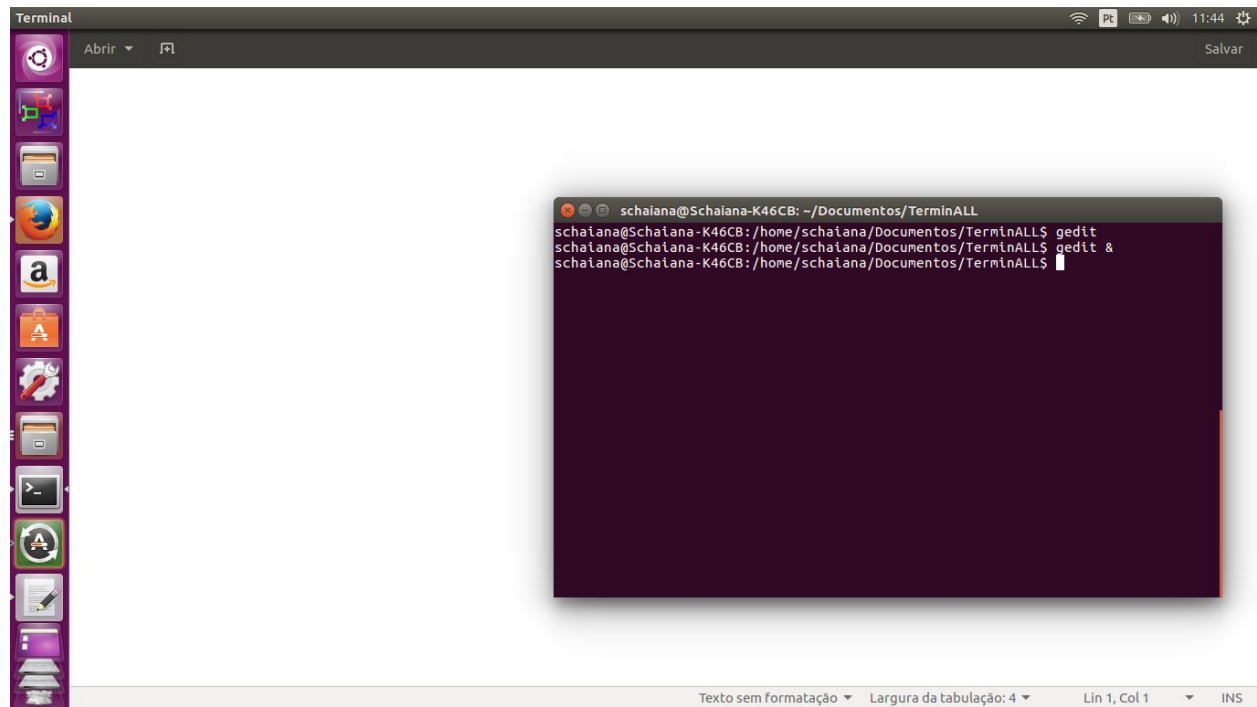
Comando com espaço depois:



```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ gedit &
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$
```

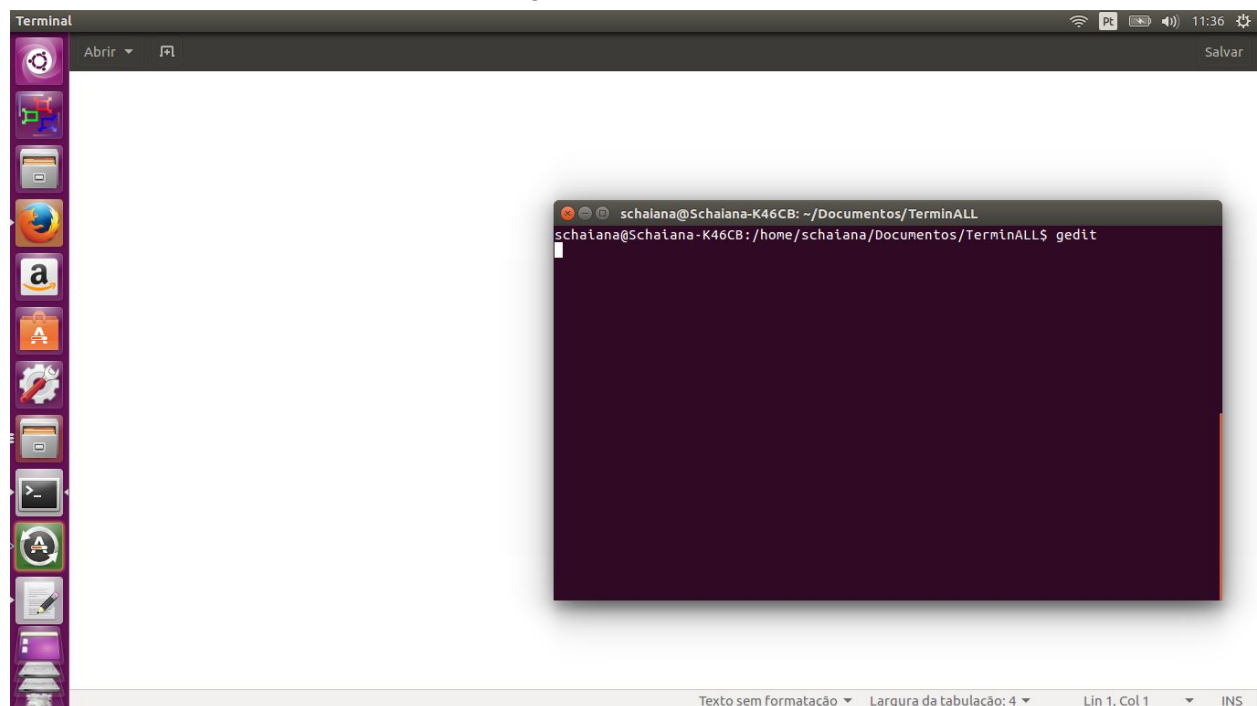
Comando com &:

Note que o processo do terminal está desbloqueado e o gedit está aberto.



Comando sem &:

Note que o terminal está bloqueado e o gedit está executando



Comandos básicos do linux (ls, ls -la, date, ifconfig):

```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ ls
lib  Makedefs  Makefile  TerminALL  TerminALL.cc  TerminALL.o
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ ls -la
total 380
drwxrwxrwx 4 schiana schiana 4096 Mar 30 15:49 .
drwxr-xr-x 3 schiana schiana 4096 Mar 29 14:24 ..
-rwxrwxrwx 1 schiana schiana 3512 Out 28 2015 .cproject
drwxrwxrwx 2 schiana schiana 4096 Mar 30 15:49 lib
-rwxrwxrwx 1 schiana schiana 563 Mar 27 16:57 Makedefs
-rwxrwxrwx 1 schiana schiana 240 Mar 27 16:57 Makefile
-rwxrwxrwx 1 schiana schiana 816 Out 29 2015 .project
drwxrwxrwx 2 schiana schiana 4096 Mar 24 14:03 .settings
-rwxrwxr-x 1 schiana schiana 313672 Mar 30 15:49 TerminALL
-rwxrwxrwx 1 schiana schiana 437 Mar 27 17:02 TerminALL.cc
-rw-rw-r-- 1 schiana schiana 33416 Mar 27 17:02 TerminALL.o
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$
```

```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ date
Qui Mar 30 16:18:38 BRT 2017
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$
```


Comando exit:

```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ exit
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$
```

Comando jobs

```
schaiana@Schaiana-K46CB: ~/Documentos/TerminALL
schaiana@Schaiana-K46CB:~/Documentos/TerminALL$ ./TerminALL
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ firefox &
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ gedit &
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$ jobs
20723 - firefox - zombie
20735 - gedit - sleeping
schaiana@Schaiana-K46CB:/home/schaiana/Documentos/TerminALL$
```

Alguns comandos foram testados e não funcionaram como no terminal do linux, pois precisam de algumas funcionalidades que não foram implementadas no projeto:

- cd
- sudo
- Up arrow

Referências:

- <http://www.cplusplus.com/> acesso em 25 de março de 2017.
- [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=SOP-EngTel_\(p%C3%A1gina\)&oldid=123789](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=SOP-EngTel_(p%C3%A1gina)&oldid=123789) acesso em 25 de março de 2017.